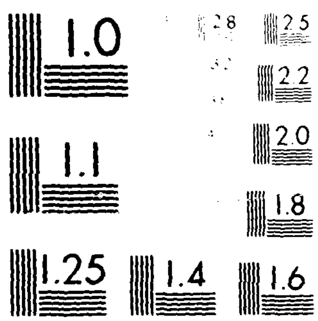
AD-A112 148 ARMY MISSILE COMMAND REDSTONE ARSENAL AL SYSTEMS SI--ETC F/G 9/2
USER'S MANUAL FOR CONTUR. A CONTOUR PLOTTING PACKAGE.(U)
AUG 81 D W ALSPAUGH, M M HALLUM
UNCLASSIFIED DRSMI/RD-81-20-TR SBI-AD-E950 212 NL

1 OF 1
AD A
112148

END
DATE
FILMED
04-82
DTIC



MICROCOPY RESOLUTION TEST CHART

(13) AD-E 950 212

TECHNICAL REPORT RD-81-20

USER'S MANUAL FOR CONTUR

A CONTOUR PLOTTING PACKAGE

Dale W. Alspaugh
School of Aeronautics and Astronautics
Purdue University
West Lafayette, Indiana 47907

Maurice M. Hallum, III
Systems Simulation and Development Directorate
US Army Missile Laboratory

August 1981

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898

Approved for public release; distribution unlimited.

DISPOSITION INSTRUCTIONS

DISCLAIMER

TRADE NAMES

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RD-81-20	2. GOVT ACCESSION NO. AD-A112 148	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* USER'S MANUAL FOR CONTUR A CONTOUR PLOTTING PACKAGE		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dale W. Alspaugh Maurice M. Hallum, III.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander, US Army Missile Command ATTN: DRSMI-RD Redstone Arsenal, AL 35898		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, US Army Missile Command ATTN: DRSMI-RPT Redstone Arsenal, AL 35898		12. REPORT DATE 19 August 1981 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computer graphics
Contour plotting
Graphics
Topography

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A program is developed which accepts arrays of X, Y, and Z coordinates as inputs and generates contours of constant values of Z. The present version is prepared for the Perkin-Elmer Interdata 8/32. The subroutines can be adapted to any other computer installation having graphics capability. The required changes are outlined in a section of the report.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

I. INTRODUCTION

A set of FORTRAN subroutines for contour plotting has been prepared. These subroutines accept arrays of X, Y, and Z coordinates as input and draw contours of constant values of Z. The present version has been prepared for use on the Interdata system located in the Systems Simulation and Development Directorate of the US Army Missile Command (MICOM) at Redstone Arsenal, Alabama. The subroutines can be adapted to any other computer installation having output graphics capability by rewriting certain subroutines which provide the commands to the output graphics devices. A later section of this documentation will outline those changes necessary to transfer to another system.

The source program for the complete set of subroutines is contained in Appendix A. THe source program is in FORTRAN.

II. OPERATION AND FEATURES

The contour plotter operates by finding the intersection of a plane parallel to the X-Y plane with a plane constructed so as to pass through three neighboring grid points. The theory of this operation is presented in the Theory Section. Since the intersection of two planes is a straight line segment, the contours are comprised of straight line segments. Clearly, the smaller the mesh, the more contour detail will be revealed. It is not necessary that the X-Y grid be uniform or even rectangular.

In normal operations the subroutines will automatically carry out scaling operations to set up appropriate scale factors. When this mode of operation is selected, the scaling will be such as to include the maximum and minimum values of X and Y.

If desired, it is possible to enable an option which causes the plot to "zoom in" to cover any desired rectangular region. In this case, the plot scales are automatically adjusted so that the desired region fills the display space on the output graphics device.

As previously noted, the grid does not have to be rectangular. It should be noted that sometimes, the use of non-rectangular grids may result in plots which display a skew distortion.

A contour numbering option is available which causes the contours to be numbered sequentially. If the numbering option is used, it is necessary to specify a parameter which identifies the number of points on a contour to be plotted between the numbering. If the number of points between numbering is given as zero, the contour will be numbered the first time that a point is plotted and not thereafter. Not all contours need be numbered.

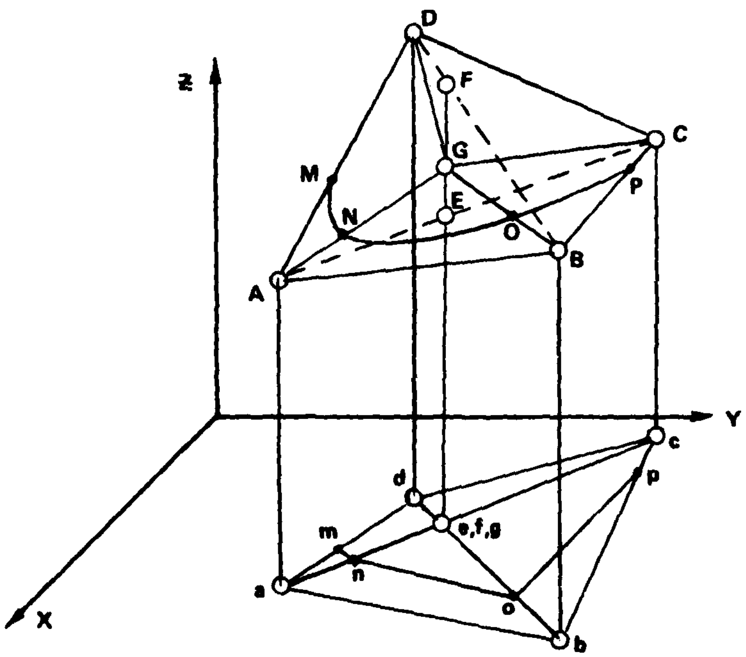
The contour plotting package is not limited in the number of grid points which may be supplied or in the number of separate value of Z for which contours may be drawn. However, it must be noted that large arrays of grid points may make the plotting process very slow. In the present version, the maximum number of contours which may be labeled is fifty.

III. THEORY OF OPERATION

The basis for operation of the contour plotter lies in several elementary geometric concepts. Specifically:

- The intersection of two planes is a straight line
- A plane can be constructed which passes through any three distinct points in space
- A straight line segment is uniquely determined by its end point

In actual operation, the plotter works by scanning through the grid in a columnar manner considering four neighboring grid points. A typical set of four neighboring grid points is shown in the figure below. This set of points forms the complex A,B,C,D. In this figure, upper case symbols represent points in the three dimensional space X, Y, Z and the lower case symbols represent the projection of the corresponding upper case point onto the X, Y plane.



In order to plot a contour, the continuous surface in the region of the complex is approximated by the four triangular plane areas (or platelets) shown in the figure as ABG, BCG, CDG, and DAG. Points e, f, and g all lie on the intersection of the diagonals ac and bd. Thus, the points E, F, and G all have the same X and Y coordinates as the intersection point g. These coordinates are easily computed from the coordinates of the projected points a, b, c and d. Thus,

$$X_G = \frac{(X_C Y_A - X_A Y_C)(X_D - X_B) - (X_D Y_B - X_B Y_D)(X_C - X_A)}{(Y_D - Y_B)(X_C - X_A) - (Y_C - Y_A)(X_D - X_B)}$$

$$Y_G = \frac{(X_A Y_C - X_C Y_A)(Y_D - Y_B) - (X_B Y_D - X_D Y_B)(Y_C - Y_A)}{(X_D - X_B)(Y_C - Y_A) - (X_C - X_A)(Y_D - Y_B)}$$

Using linear interpolation, the Z coordinates of points E and F are computed as:

$$Z_E = Z_A + (Z_C - Z_A)\,\frac{(X_G - X_A)^2 + (Y_G - Y_A)^2}{(X_C - X_A)^2 + (Y_C - Y_A)^2}$$

$$Z_F = Z_B + (Z_D - Z_B)\,\frac{(X_G - X_B)^2 + (Y_G - Y_B)^2}{(X_D - X_B)^2 + (Y_D - Y_B)^2}$$

The Z coordinate of G is chosen to be the mean of Z_E and Z_F, i.e.,

$$Z_G = \frac{(Z_E + Z_F)}{2}$$

Each of the four platelets (e.g., ABG) is examined to determine if a plane parallel to and a distance Z above the XY plane will intersect with that platelet. In order to make this determination, the Z values of the vertices of the platelet are placed in ascending order. The desired Z value is then compared to the largest and smallest values of the vertex values. If the desired value lies within the range, the plane will intersect the platelet. Further, if the desired contour value is greater than the intermediate point Z value, the contour plane will intersect the edges of the platelet which connect the vertex having the maximum Z value with the vertex of minimum value and with the vertex of intermediate value. If the contour value is less than the intermediate point Z value, the contour plane will intersect the edges of the platelet which connect the vertex having the maximum Z value with the vertex of minimum value and with the vertex of intermediate value. If the contour value is less than the intermediate point Z value, the contour plane will intersect the edges of the platelet which connect the vertex of minimum Z value with the vertex of maximum value and with the vertex of intermediate value.

The coordinates of the edge intersection points are then determined by a linear interpolation. Let $(X_{max}, Y_{max}, Z_{max})$, $(X_{int}, Y_{int}, Z_{int})$, and $(X_{min}, Y_{min}, Z_{min})$ be the ordered sequence of vertices of a platelet. If $Z > Z_{int}$ then,

$$X_i = X_{max} + \frac{(Z - Z_{max})}{(Z_{int} - Z_{max})} (X_{int} - X_{max})$$

$$Y_i = Y_{max} + \frac{(Z - Z_{max})}{(Z_{int} - Z_{max})} (Y_{int} - Y_{max})$$

$$X_j = X_{max} + \frac{(Z - Z_{max})}{(Z_{min} - Z_{max})} (X_{min} - X_{max})$$

$$Y_j = Y_{max} + \frac{(Z - Z_{max})}{(Z_{min} - Z_{max})} (Y_{min} - Y_{max}).$$

If $Z < Z_{int}$ then

$$X_i = X_{min} + \frac{(Z - Z_{min})}{(Z_{int} - Z_{min})} (X_{int} - X_{min})$$

$$Y_i = Y_{min} + \frac{(Z - Z_{min})}{(Z_{int} - Z_{min})} (Y_{int} - Y_{min})$$

$$X_j = X_{min} + \frac{(Z - Z_{min})}{(Z_{max} - Z_{min})} (X_{max} - X_{min})$$

$$Y_j = Y_{min} + \frac{(Z - Z_{min})}{(Z_{max} - Z_{min})} (Y_{max} - Y_{min}).$$

A line is then drawn from the point (X_i, Y_i) to (X_j, Y_j).

This process is carried out for all four of the platelets at the end of this cycle. At this point a portion of the contour is complete as shown by the line mnop in the figure. The entire process is repeated for each set of four neighboring points.

It should be noted that a somewhat similar algorithm could be designed using only two platelets to represent the surface. In this case, a diagonal would be drawn between two of the diagonal points. For example, the platelets might be ABC and ACD. This choice of platelets sometimes results in a skewing of the contours. The four platelet scheme essentially eliminates this skewing. To the best of the author's knowledge, the four platelet scheme has not been previously described.

IV. CALLING SEQUENCE

To utilize the package as a contour plotter, the following FORTRAN statement must be used:

```
CALL SLICE (ZV,X,Y,Z,M,N,MM,NN,NZ,MS,ZOOM,XLO,XUP,YLO,YUP,TYPE,TAG,
NTAG,TAGV,NTIMES).
```

The arguments in this call have the following meanings and characteristics.

Argument	Type	Dimension[1]	Definition
ZV	REAL	NZ	The Z values of the contour to be drawn
X	REAL	MMXNN	The X values of the grid points
Y	REAL	MMXNN	The Y values of the grid points
Z	REAL	MMXNN	The Z values of the grid points
M	INTEGER		The number of "columns" in the X-Y grid
N	INTEGER	M	The number of grid points in each "column"[2]
MM	INTEGER		The maximum number of "columns" in the grid[3]
NN	INTEGER		The maximum number of "rows" in the grid[3]
NZ	INTEGER		The number of contours to be drawn
MS	INTEGER		If the zoom option is not selected and if MS=1, automatic scaling of the grid will occur.
ZOOM	LOGICAL		If ZOOM is true, the ZOOM option is selected
XLO	REAL		Minimum X value of the plotting window
XUP	REAL		Maximum X value of the plotting window
YLO	REAL		Minimum Y value of the plotting window

[1]If not a simple variable.
[2]Each "column" need not have the same number of points.
[3]Variable dimension statements are utilized for several variables.

Argument	Type	Dimension[1]	Definition
YUP	REAL		Maximum Y value of the plotting window
TYPE	INTEGER	NZ	Sets the type of line to be drawn for each contour. Permissible values are 'LINE', 'DASH', 'DOT'
TAG	LOGICAL		If TAG is true, the contours specified in TAGV will be labeled with their contour number the first time a point is plotted in the window and every NTIMES points thereafter.
NTAG	INTEGER		The number of contours to be tagged
TAGV	INTEGER	NTAG	A vector containing the contour numbers of the contours to be tagged[2]
NTIMES	INTEGER		The number of points plotted within the plotting window between labeling of the contours. If NTIMES = 0, the contours will be tagged at the first point plotted only.

The first use of subroutine SLICE or any of the other subroutines which utilize output graphic pen motion must be preceeded by the following two calls:

```
CALL INIT

CALL DEVICE (NDV)³
```

Successive calls to SLICE should be separated by

```
CALL PAGE
```

which erases the screen. A call to subroutine COPY

```
CALL COPY
```

causes a hard copy to be produced.

---

[1]If not a simple variable
[2]Not all contours need be tagged if the tag option is selected.
[3]The device number NDV is dependent upon the output graphic device being used:
NDV = 1 for HP2648 Terminals
NDV = 2 for TEK4006 and TEK4051 Terminals
NDV = 3 for TEK4014 Terminals
NDV = 4 for TEK 4662 Flat Bed Plotter

If this package is to be adapted to another computer system or graphics package, it will be necessary to rewrite certain subroutines. These subroutines provide basic graphics operations such as pen motion, etc. In the paragraphs to follow, these subroutines and their functions are described.

SUBROUTINE UWINDO (XLO,XUP,YLO,YUP)

This subroutine should set the upper and lower bounds on the X and Y coordinates of the display window. These values are respectively XUP, XLO, YUP, YLO. Note that these values are in virtual units, i.e., number being plotted, and are not meant to be physical distance units on the display.

SUBROUTINE UMOVE (X,Y)

This subroutine causes the pen to move from its current location to a point having virtual coordinates X,Y. This pen movement is accomplished with the pen "up", i.e., no line is drawn.

SUBROUTINE UPEN (X,Y)

This subroutine causes a line to be drawn from the current pen position to the point having virtual coordinates X,Y. If any portion of this line occurs outside of the display window, that portion of the line should be visible.

SUBROUTINE UPRNT1(N,B)

This subroutine causes an integer number N to be printed at the current pen location. If that point is outside of the display window, the printing should be invisible. The argument B is a dummy.

SUBROUTINE USET(A)

This subroutine is used to define the line type that is to be drawn for all subsequent calls of UPEN until changed by another call of USET. If A = 'DASH', the line is a dashed line. If A = 'LINE' a solid line is drawn. If A = 'DOT' a dotted line is drawn.

APPENDIX A

```
$BATCH
      SUBROUTINE SLICE(ZV,XX,YY,ZZ,M,N,MM,NN,NZ,MS,ZOOM,XLO,XUP
      YLO,YUP,TYPE,TAG,NTAG,TAGV,NTIMES)
      LOGICAL ZOOM,TAG,NEWTAG
      DIMENSION XX(MM,NN),YY(MM,NN),ZZ(MM,NN),N(M),X(3),Y(3),Z(3)
      ,XP(3),YP(3),ZP(3),ZV(NZ),TX(4),TY(4),TZ(4),NEWTAG(50)
      INTEGER TYPE(NZ),TAGV(NTAG),MTAG(50)
*****************************************************************
*     THIS SUBROUTINE DRAWS A CONTOUR MAP FOR THE Z COMPONENTS    *
*     CORRESPONDING TO A A SET OF MESH POINTS. THE MESH DOES NOT  *
*     HAVE TO BE RECTANGULAR. IF IT IS DESIRED TO SET UP APPROPRIATE *
*     SCALE FACTORS, MS SHOULD BE 1                               *
*     ZV IS AN ARRAY CONTAINING THE VALUES OF Z FOR WHICH CONTOURS *
*     ARE DESIRED                                                 *
*     IF ZOOM=.TRUE. , THE FIELD OF VIEW WILL BE ZOOMED IN ON THE *
*     RECTANGULAR AREA SPECIFIED BY XLO,XUO,YLO,YUP. IF .NOT.ZOOM *
*     XLO,XUP,YLO,YUP WILL RETURN THE !NEAT! VALUES COMPUTED FOR  *
*     CSCALE                                                      *
*     MM AND NN ARE THE MAXIMUM DIMENSIONS OF THE DOUBLE SUBSCRIPTED *
*     VARIABLES AS GIVEN IN THE ORIGINAL ABSOLUTE DIMENSION STATEMENT*
*     THE INTEGER VECTOR !TYPE! IS USED TO SET THE LINE TYPE FOR EACH*
*     CONTOUR, IF THE LINE TYPES ARE NOT SPECIFIED, THEY DEFAULT TO *
*     SOLID LINES. THE DASH LENGTH AND SPACE LENGTH ARE SET AT .0366 *
*     INCHES. TYPES AVAILABLE INCLUDE 'LINE','DOT,'DASH'.         *
*     IF 'TAG'=.TRUE. , THE CONTOURS SPECIFIED IN TAGV WILL BE TAGGED*
*     WITH THEIR CONTOUR NUMBER THE FIRST TIME THE CONTOUR APPEARS *
*     IN THE WINDOW. THE NUMBER OF CONTOURS TO BE TAGGED IS NTAG  *
*     THE MAXIMUM NUMBER OF CURVES THAT CAN BE USED WHEN THE TAG  *
*     OPTION IS SELECTED IS 50.                                   *
*     THE TAGGED CONTOURS WILL BE MARKED THE FIRST TIME THAT A    *
*     POINT ON THAT CONTOUR IS PLOTTED WITHIN THE WINDOW AND EVERY *
*     NTIMES TIMES THERAFTER.                                     *
*     VERSION 7.6                    7/10/80                      *
*****************************************************************
      IF(.NOT.TAG) GO TO 153
      DELX=.005*ABS(XUP-XLO)
      DELY=.0075*ABS(YUP-YLO)
      DO 151 I=1,NZ
      MTAG(I)=1
 151  NEWTAG(I)=.FALSE.
      DO 152 I=1,NTAG
 152  NEWTAG(TAGV(I))=.TRUE.
 153  CONTINUE
      DO 150 I=1,NZ
      IF(TYPE(I).NE.'LINE'.AND.TYPE(I).NE.'DASH'.AND.TYPE(I).NE.'DOT')
       TYPE(I)='LINE'
 150  CONTINUE
      CALL UPSET('SETDASH',12.)
      IF(.NOT.ZOOM.AND.MS.EQ.1) CALL CSCALE(XX,YY,M,N,MM,NN,XLO,XUP,
      YLO,YUP)
      IF(ZOOM) CALL WINDOW(XLO,XUP,YLO,YUP)
      MMAX=M-1
      DO 100 I=1,MMAX
      NMAX=N(I)-1
      IF(N(I+1).LT.N(I)) NMAX=N(I+1)-1
      DO 100 J=1,NMAX
      IF(.NOT.ZOOM) GO TO 510
      XU=XX(I,J)
      YU=YY(I,J)
      XL=XX(I,J)
      YL=YY(I,J)
      DO 500 K1=1,2
```

```
      DO 500 K2=1,2
      KI=I+K1-1
      KJ=J+K2-1
      IF(XX(KI,KJ).GT.XG) XG=XX(KI,KJ)
      IF(XX(KI,KJ).GT.XU) XU=XX(KI,KJ)
      IF(YY(KI,KJ).GT.YG) YG=YY(KI,KJ)
500   IF(YY(KI,KJ).GT.YU) YU=YY(KI,KJ)
      IF(XG.GT.XLO.OR.XU.GT.XUP.OR.YG.LT.YLO.OR.YL.GT.YUP)
      GO TO 100
510   TX(1)=XX(I,J)
      TX(2)=XX(I+1,J)
      TX(3)=XX(I+1,J+1)
      TX(4)=XX(I,J+1)
      TY(1)=YY(I,J)
      TY(2)=YY(I+1,J)
      TY(3)=YY(I+1,J+1)
      TY(4)=YY(I,J+1)
      TZ(1)=ZZ(I,J)
      TZ(2)=ZZ(I+1,J)
      TZ(3)=ZZ(I+1,J+1)
      TZ(4)=ZZ(I,J+1)
      X42=TX(4)-TX(2)
      X31=TX(3)-TX(1)
      Y42=TY(4)-TY(2)
      Y31=TY(3)-TY(1)
      DENOM=Y42*X31-Y31*X42
      XC=((TX(3)*TY(1)-TX(1)*TY(3))*X42-(TX(4)*TY(2)-TX(2)*TY(4))
     *X31)/DENOM
      YC=((TX(2)*TY(4)-TX(4)*TY(2))*Y31-(TX(1)*TY(3)-TX(3)*TY(1))
     *Y42)/DENOM
      ZC1=TZ(1)+(TZ(3)-TZ(1))*SQRT(((XC-TX(1))**2+(YC-TY(1))**2)/
     (X31*X31+Y31*Y31))
      ZC2=TZ(2)+(TZ(4)-TZ(2))*SQRT(((XC-TX(2))**2+(YC-TY(2))**2)/
     (X42*X42+Y42*Y42))
      ZC=(ZC1+ZC2)/2.
      Z(1)=ZC
      X(1)=XC
      Y(1)=YC
      DO 400 L=1,4
      Z(2)=TZ(L)
      X(2)=TX(L)
      Y(2)=TY(L)
      IF(L.EQ.4) GO TO 401
      Z(3)=TZ(L+1)
      X(3)=TX(L+1)
      Y(3)=TY(L+1)
      GO TO 402
401   Z(3)=TZ(1)
      X(3)=TX(1)
      Y(3)=TY(1)
402   CALL SORT3(X,Y,Z,XP,YP,ZP)
      DO 200 K=1,NZ
      IF(TYPE(K).EQ.'DOT') CALL UPSET('SETDASH',92.)
      CALL INTER(ZV(K),XP,YP,ZP,TYPE(K),TAG,NEWTAG(K),K,NTIMES,NTAG(K)
     ,DELX,DELY,XLO,XUP,YLO,YUP,ZOOM)
      IF(TYPE(K).EQ.'DOT') CALL UPSET('SETDASH',12.)
200   CONTINUE
400   CONTINUE
100   CONTINUE
      RETURN
      END
      SUBROUTINE INTER(ZV,X,Y,Z,TYPE,TAG,NEWTAG,NTAG,NPTS,MPTS,
     DELX,DELY,XLO,XUP,YLO,YUP,ZOOM)
```

```
      LOGICAL TAG,NEWTAG,KLIP,ZOOM
      INTEGER TYPE
      DIMENSION X(3),Y(3),Z(3)
**********************************************************************
*     THIS SUBROUTINE IS CALLED BY SLICE AND DRAWS THE LINE REPRESENT*
*     ING THE INTERSECTION OF A HORIZONTAL PLANE AND THE TRIANGULAR  *
*     PLANE REGION                                                   *
*     VERSION 3.3                    7/08/80                         *
**********************************************************************
      KLIP=.FALSE.
      IF(ZV.LT.Z(1).OR.ZV.GT.Z(3)) GO TO 100
      IF(Z(3).LE.Z(1)) GO TO 100
      IF(TYPE.NE.'DOT') CALL USET(TYPE)
      IF(TYPE.EQ.'DOT') CALL USET('DASH')
      IF(Z(3).LE.Z(2)) GO TO 210
      IF(Z(2).LE.Z(1)) GO TO 200
 203  X4=((Z(3)-ZV)*X(1)+(ZV-Z(1))*X(3))/(Z(3)-Z(1))
      Y4=((Z(3)-ZV)*Y(1)+(ZV-Z(1))*Y(3))/(Z(3)-Z(1))
      IF(TAG.AND.NEWTAG) GO TO 250
 252  CALL UMOVE(X4,Y4)
      IF(ZV.GT.Z(2)) GO TO 20
      X5=((Z(2)-ZV)*X(1)+(ZV-Z(1))*X(2))/(Z(2)-Z(1))
      Y5=((Z(2)-ZV)*Y(1)+(ZV-Z(1))*Y(2))/(Z(2)-Z(1))
      GO TO 30
 20   X5=((Z(3)-ZV)*X(2)+(ZV-Z(2))*X(3))/(Z(3)-Z(2))
      Y5=((Z(3)-ZV)*Y(2)+(ZV-Z(2))*Y(3))/(Z(3)-Z(2))
 30   IF(ZOOM) CALL CLIP(X4,Y4,X5,Y5,XLO,XUP,YLO,YUP,KLIP)
      IF(.NOT.KLIP) CALL UPEN(X5,Y5)
 100  CONTINUE
      RETURN
 200  IF(ZV.LE.Z(1)) GO TO 201
      GO TO 203
 210  IF(ZV.GE.Z(3)) GO TO 301
      GO TO 203
 201  X4=X(1)
      Y4=Y(1)
      X5=X(2)
      Y5=Y(2)
      GO TO 202
 301  X4=X(2)
      Y4=Y(2)
      X5=X(3)
      Y5=Y(3)
 202  CALL UMOVE(X4,Y4)
      IF(ZOOM) CALL CLIP(X4,Y4,X5,Y5,XLO,XUP,YLO,YUP,KLIP)
      IF(.NOT.KLIP) CALL UPEN(X5,Y5)
      RETURN
 250  IF(X4.GE.XLO.AND.X4.LE.XUP.AND.Y4.GE.YLO.AND.Y4.LE.YUP)
       GO TO 251
      GO TO 252
 251  IF(MPTS.EQ.1.OR.MPTS.EQ.NPTS) GO TO 253
      MPTS=MPTS+1
      GO TO 252
 253  MPTS=2
      IF(NPTS.EQ.0) NEWTAG=.FALSE.
      CALL USET('LINE')
      CALL UMOVE(X4,Y4-DELY)
      X6=X4
      Y6=Y4-DELY
      X7=X4
      Y7=Y4+DELY
      CALL CLIP(X6,Y6,X7,Y7,XLO,XUP,YLO,YUP,KLIP)
      IF(.NOT.KLIP) CALL UPEN(X7,Y7)
```

```
      CALL UMOVE(X4-DELX,Y4)
      X6=X4-DELX
      Y7=Y4
      Y6=Y4
      X7=X4+DELX
      CALL CLIP(X6,Y6,X7,Y7,XLO,XUP,YLO,YUP,KLIP)
      IF(.NOT.KLIP) CALL UPEN(X7,Y7)
      CALL UPRNT1(NTAG,'INTEGERNUMBER')
      IF(TYPE.EQ.'DOT'.OR.TYPE.EQ.'DASH') CALL USET('DASH')
      IF(TYPE.EQ.'DASH') CALL UPSET('SETDASH',12.)
      IF(TYPE.EQ.'DOT') CALL UPSET('SETDASH',92.)
      GO TO 252
      END
      SUBROUTINE SORT3(XX,YY,ZZ,X,Y,Z)
      DIMENSION XX(3),YY(3),ZZ(3),X(3),Y(3),Z(3)
*********************************************************************
*     THIS SUBROUTINE IS CALLED BY INTER AND SORTS A SET OF THREE   *
*     POINTS INTO ASCENDING ORDER BASED ON THE Z VALUES             *
*     VERSION 1.0               5/16/80                             *
*********************************************************************
      DO 31 I=1,3
      X(I)=XX(I)
      Y(I)=YY(I)
 31   Z(I)=ZZ(I)
      DO 2 J=1,2
      DO 1 I=1,2
      IF(Z(I).LT.Z(I+1)) GO TO 1
      DUM=Z(I)
      DUMX=X(I)
      DUMY=Y(I)
      Z(I)=Z(I+1)
      Y(I)=Y(I+1)
      X(I)=X(I+1)
      Z(I+1)=DUM
      X(I+1)=DUMX
      Y(I+1)=DUMY
 1    CONTINUE
 2    IF(Z(1).LE.Z(2).AND.Z(2).LE.Z(3)) GO TO 3
      WRITE(6,4)X(1),Y(1),Z(1),X(2),Y(2),Z(2),X(3),Y(3),Z(3)
 4    FORMAT(//,' ERROR DETECTED IN SUBROUTINE SORT'//3(3(2X,E15.8)/))
 3    RETURN
      END
      SUBROUTINE CSCALE(X,Y,M,N,MM,NN,XLOW,XUP,YLOW,YUP)
      DIMENSION X(MM,NN),Y(MM,NN),N(M)
*********************************************************************
*     THIS SUBROUTINE IS CALLED BY SLICE AND SETS UP THE PLOTTING   *
*     WINDOW IN VIRTUAL SPACE                                       *
*     VERSION 3.0.                          6/24/80                  *
*********************************************************************
      YMAX=-1.E60
      DO 1 I=1,M
      IF(Y(I,N(I)).GT.YMAX) YMAX=Y(I,N(I))
 1    CONTINUE
      YMIN=YMAX
      DO 2 I=1,M
      IF(Y(I,1).LT.YMIN) YMIN=Y(I,1)
 2    CONTINUE
      XMIN=X(1,1)
      XMAX=X(M,1)
      CALL NEAT(XMAX,XMIN,XUP,XLOW)
      CALL NEAT(YMAX,YMIN,YUP,YLOW)
      CALL UWINDO(XLOW,XUP,YLOW,YUP)
      RETURN
```

```
      END
      SUBROUTINE NEAT(XMAX,XMIN,XUP,XLOW)
************************************************************************
*     THIS SUBROUTINE IS CALLED BY CSCALE AND DETERMINES INEATI VALUES
*     FOR PLOTTING SCALES                                              *
*     VERSION 1.0                            5/16/80                   *
************************************************************************
      IF(ABS(XMAX).LT.ABS(XMIN)) GO TO 1
      EX=AINT(ALOG10(ABS(XMAX)))
 2    B=10.**EX
      XUP=(AINT(XMAX/B)+1.)*B
      IF(INT(XMAX/B).EQ.0) XUP=0.
      XLOW=(AINT(XMIN/B)-1.)*B
      IF(INT(XMIN/B).EQ.0) XLOW=0.
      RETURN
 1    EX=AINT(ALOG10(ABS(XMIN)))
      GO TO 2
      END
      SUBROUTINE DWINDO(XLO,XUP,YLO,YUP)
      DIMENSION WIND(15)
      COMMON /BLK2/ XSCALE,XZER,YSCALE,YZER
      COMMON /BLK5/ WIND
      XSCALE=(WIND(2)-WIND(1))/(XUP-XLO)
      YSCALE=(WIND(4)-WIND(3))/(YUP-YLO)
      XZER=WIND(1)-XSCALE*XLO
      YZER=WIND(3)-YSCALE*YLO
      RETURN
      END
      SUBROUTINE [illegible](X,Y)
      CALL GRAPH
      CALL PLOT(X,Y,7,1,1)
      RETURN
      END
      SUBROUTINE [illegible](X,Y)
      CALL PLOT(X,Y,7,1,1)
      CALL ALPHA
      RETURN
      END
      SUBROUTINE [illegible](A,B)
      COMMON /BLK1/ [illegible],IFC,LU
      DIMENSION PAGE(5)
      CALL ALPHA
      CALL OUT
      WRITE(LU,100) A
100   FORMAT(I2)
      RETURN
      END
      SUBROUTINE [illegible](A,B)
      RETURN
      END
      SUBROUTINE [illegible](A)
      IF(A.EQ.'[illegible]') CALL [illegible]TYPE(4)
      IF(A.EQ.'[illegible]') CALL [illegible]TYPE(2)
      IF(A.EQ.'[illegible]') CALL [illegible]TYPE(1)
      RETURN
      END
      SUBROUTINE CLIP(X4,Y4,X5,Y5,XLO,XUP,YLO,YUP,KLIP)
      LOGICAL KLIP
      KLIP=.FALSE.
      IF(X4.GE.XLO.AND.X4.LE.XUP.AND.X5.GE.XLO.AND.X5.LE.XUP.
     $AND.Y4.GE.YLO.AND.Y4.LE.YUP.AND.Y5.GE.YLO.AND.Y5.LE.YUP)
     & RETURN
 1    IF(X4.LT.XLO.AND.X5.LT.XLO) GO TO 999
```

```
      IF(X4.GT.XUP.AND.X5.GT.XUP) GO TO 999
      IF(Y4.LT.YLO.AND.Y5.LT.YLO) GO TO 999
      IF(Y4.GT.YUP.AND.Y5.GT.YUP) GO TO 999
      IF(X4.LT.XLO.AND.X5.GT.XLO) GO TO 10
      IF(X4.GT.XUP.AND.X5.LT.XUP) GO TO 20
      IF(X5.LT.XLO.AND.X4.GT.XLO) GO TO 30
      IF(X5.GT.XUP.AND.X4.LT.XUP) GO TO 40
      IF(Y4.LT.YLO.AND.Y5.GT.YLO) GO TO 50
      IF(Y4.GT.YUP.AND.Y5.LT.YUP) GO TO 60
      IF(Y5.LT.YLO.AND.Y4.GT.YLO) GO TO 70
      IF(Y5.GT.YUP.AND.Y4.LT.YUP) GO TO 80
      GO TO 200
999   KLIP=.TRUE.
      RETURN
10    XT=XLO
      YT=Y4+(Y5-Y4)*(XLO-X4)/(X5-X4)
      X4=XT
      Y4=YT
      GO TO 100
20    XT=XUP
      YT=Y4+(Y5-Y4)*(X4-XUP)/(X4-X5)
      X4=XT
      Y4=YT
      GO TO 100
30    XT=XLO
      YT=Y5+(Y4-Y5)*(XLO-X5)/(X4-X5)
      X5=XT
      Y5=YT
      GO TO 100
40    XT=XUP
      YT=Y5+(Y4-Y5)*(X5-XUP)/(X5-X4)
      X5=XT
      Y5=YT
      GO TO 100
50    YT=YLO
      XT=X4+(X5-X4)*(YLO-Y4)/(Y5-Y4)
      X4=XT
      Y4=YT
      GO TO 100
60    YT=YUP
      XT=X4+(X5-X4)*(Y4-YUP)/(Y4-Y5)
      X4=XT
      Y4=YT
      GO TO 100
70    YT=YLO
      XT=X5+(X4-X5)*(YLO-Y5)/(Y4-Y5)
      X5=XT
      Y5=YT
      GO TO 100
80    YT=YUP
      XT=X5+(X4-X5)*(Y5-YUP)/(Y5-Y4)
      X5=XT
      Y5=YT
100   IF(X4.GE.XLO.AND.X4.LE.XUP.AND.Y4.GE.YLO
     &AND.Y4.LE.YUP.AND.X5.GE.XLO.AND.X5.LE.XU
     &AND.Y5.GE.YLO.AND.Y5.LE.YUP) GO TO 200
      GO TO 1
200   CALL UMOVE(X4,Y4)
      RETURN
      END
$BEND
```

DISTRIBUTION

	No. of Copies
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, IL 60616	1
DRSMI-LP, Mr. Voigt	1
-RPR	15
-RPT, Record Copy	1
Reference Copy	1
-RDF	20
-RDR	1
-RDF, Mr. Brazzel	1
Mr. Landingham	1

ED

8